

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

INTELIGENTNÍ DŮM S JEDNOČIPOVÝM POČÍTAČEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ŠIMON PIRO

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

INTELIGENTNÍ DŮM S JEDNOČIPOVÝM POČÍTAČEM

SMART HOME WITH SINGLE BOARD COMPUTER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ŠIMON PIRO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MARTIN KOUTNÝ

BRNO 2011



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Šimon Piro

ID: 115256

Ročník: 3

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Inteligentní dům s jednočipovým počítačem

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte současné koncepce inteligentních domů. Ve své práci se zaměřte na jejich nedostatky. Realizujte návrh nové koncepce Inteligentního domu pomocí sběrnice RS-485, PLC nebo Zigbee. V rámci bakalářské práce realizujte grafickou demonstrační aplikaci pro ovládání sériového rozhraní v operačním systému linux. Pro praktickou ukázkou aplikace připojte k seriovému rozhraní vhodné měřicí zařízení s kterým budete komunikovat.

DOPORUČENÁ LITERATURA:

- [1] GOODWIN, Steven. Smart Home Automation with Linux (Expert's Voice in Linux). : Apress, 2010. 312 s. ISBN 978-1430227786
- [2] HW Server. Domáci automatizace - komunikace po síťovém rozvodu 230V. HW Server [online]. 2003, 2, [cit. 2010-10-12]. Dostupný z WWW:
<<http://hw.cz/Teorie-a-praxe/Konstrukce/ART46-Domaci-automatizace---komunikace-po-sitovem-rozvodu-230V.html>>.

Termín zadání: 7.2.2011

Termín odevzdání: 2.6.2011

Vedoucí práce: Ing. Martin Koutný

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá problematikou inteligentních domů, přehledem řešení, které se v současnosti využívají, návrhem jednoduššího a finančně nenáročného řešení, výběrem řídicí jednotky v podobě jednočipového počítače s instalovaným OS Linux. V práci je dále řešena realizace grafické uživatelské aplikace pro dotykový displej, implementace vzdáleného přístupu pomocí webové HTTP stránky a návrhem periferní jednotky s integrovaným mikroprocesorem.

KLÚČOVÉ SLOVÁ

Inteligentní dům, jednočipový počítač

ABSTRACT

This thesis describes the issue of smart homes, reviews of current solutions, design simpler and less financially demanding solutions, the selection of the control unit in the form of single board computer based on an embedded Linux OS, realization of graphical user application for touch screen, implementing remote access via HTTP webpage, proposal of peripheral unit with integrated microprocessor.

KEYWORDS

Smart Home, Single Board Computer

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Inteligentný dom s jednočipovým počítačem“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

OBSAH

Úvod	9
1 Centralizácia systémov v domoch	11
1.1 Multimédia	11
1.2 Kúrenie	11
1.3 Inteligentné domy	12
1.3.1 Káblové systémy	13
1.3.2 Bezdrôtové systémy	14
1.3.3 Najpoužívanjšie systémy inteligentných domov	15
2 Realizácia riadiaceho systému	16
2.1 Výber riadiacej jednotky	16
2.1.1 SBC(Single-Board Computer)	16
2.1.2 Procesory ARM	17
2.1.3 Riadiace jednotky	18
2.2 Inicializácia riadiacej jednotky	21
2.2.1 Postup inicializácie	21
2.2.2 Bootovanie zo siete	25
2.3 Návrh systému	27
2.3.1 Návrh aplikácie pre riadiacu jednotku	28
2.4 Praktická realizácia	30
2.4.1 Realizácia aplikácie pre riadiacu jednotku	30
2.4.2 Realizácia vzdialeného prístupu	35
2.4.3 Periférna jednotka	37
3 Záver	41
Literatúra	43
Zoznam symbolov, veličín a skratiek	45
A Diagram udalostí	47

B	Schéma zapojenia teplomera	48
C	Schéma plošného spoja teplomera	49

ZOZNAM OBRÁZKOV

2.1	SBC SAM9M10	18
2.2	SBC SAM3U	19
2.3	SBC MCB2929	19
2.4	SBC6410	20
2.5	Vytváranie bootovacej SD karty	22
2.6	Nastavenie prepínačov do polohy pre bootovanie z SD karty	22
2.7	Nastavenie pripojenia v Secure CRT	23
2.8	Nastavenie USB pripojenia v DNW	23
2.9	Vypálenie U-Boot do NAND flash	24
2.10	Odoslanie image súboru pomocou DNW	24
2.11	Nastavenie prepínačov do polohy pre bootovanie z NAND Flash	25
2.12	Bloková schéma systému	27
2.13	Hlavné menu aplikácie	28
2.14	Ovládacie menu domu	28
2.15	Menu výberu izby	29
2.16	Ovládanie konkrétnej izby	29
2.17	Štruktúra XML súboru.	30
2.18	Prvý frame v QT Creatore.	31
2.19	Resources v QT Creatore.	32
2.20	Nastavenie cesty k nástroju QMake-Qt4.	35
2.21	Formulár na našej PHP stránke	36
A.1	Diagram udalostí po stlačení akčného tlačítka.	47
B.1	Schéma zapojenia teplomera.	48
C.1	Doska plošného spoja teplomera.	49

ÚVOD

V dnešnej dobe je trendom pripájať všetko do siete. To znamená pripojiť niečo niekam za účelom ich vzájomnej komunikácie. Touto sieťou je najčastejšie internet. Do internetu sú pripojené milióny zariadení, ktoré spolu komunikujú a vymieňajú si informácie. Do siete sa pripájajú aj samotní ľudia. Sú to tzv. sociálne siete (Facebook a pod.). Dôvod však je stále ten istý. Mať informácie centralizované a vždy po ruke. Táto centralizácia sa však využíva aj v menších mierkach, ako sú celosvetové mierky. Centralizujú sa štátne informácie a databázy, podnikové systémy a pod. My sme sa zamerali na ešte menšie mierky, a to centralizovanie domácností.

Aj domácnosť sa dá centralizovať. Môžeme dosiahnuť, aby spolu jednotlivé prvky v domácnosti komunikovali. Centralizáciu domov môžeme pozorovať už dlhšiu dobu. Centralizované už sú napríklad tepelné systémy a vykurovanie. Každý má doma termostat, ktorý sníma teplotu v byte a zapína a vypína kotol. Ďalší príklad takejto centralizácie sme mohli pozorovať pri príchode internetu. Internetový poskytovateľ nám privedie internetovú prípojku až domov. Neskôr už doma nestačil len jeden počítač. Pribudol server, ktorý sa stal centrálnou jednotkou multimediálnych dát. Takto vznikli domáce počítačové siete a ďalší centralizovaný prvok domácnosti.

Ak rozmýšľame nad inteligentným domom, musíme najprv zadať, čo si pod týmto pojmom predstavujeme. Musíme definovať, čo robí inteligentný dom inteligentným. V prvom rade je to dom, ktorý spĺňa požiadavky svojho užívateľa, uľahčuje a spríjemňuje mu život. Preto nesmie vybavenosť technickými prostriedkami zvýšiť čas na obsluhu, nesmie vyžadovať prílišnú technickú zdatnosť alebo nadmieru zasahovať do súkromia človeka. Až keď docielime rovnováhu týchto systémov, budeme môcť nazvať dom „inteligentným“.

V práci si postupne prejdeme jednotlivé spôsoby centralizácie. Aké riešenia sú na dnešnom trhu ponúkané, a aké spôsoby a technológie sú na to využívané. Spravíme jednoduchý návrh systému do takéhoto domu. Vyberieme riadiacu jednotku s jej následnou inicializáciou. Spravíme návrh a následnú realizáciu jednoduchej grafickej aplikácie do tejto riadiacej jednotky, aby bola čo najjednoduchšie ovládateľná dotykovým displejom. A keďže bude umiestnená v obytných priestoroch, tak by bolo

vhodné, aby bola využiteľná aj inak, ako na ovládanie domu.

Niekedy sa človek dostane do situácie, v ktorej si nie je istý, či nenechal náhodou zapnutý nejaký spotrebič alebo svetlo, resp. chce nejaký spotrebič na diaľku zapnúť. Kvôli tomu umožníme vzdialený prístup k ovládaniu domu, a to pomocou webovej stránky v jazyku PHP.

Navrhujeme jednoduchú periférnu jednotku vybavenú mikroprocesorom ATMega8 schopnú posielat' dáta po sériovej linke RS-232, a tieto dáta následne zapisovať do súboru. Tento súbor je v podstate databáza všetkých zariadení v dome a ich momentálneho stavu.

1 CENTRALIZÁCIA SYSTÉMOV V DOMOCH

Centralizácia je snaha ovládať všetko z jedného miesta. Ako narastá počet elektrických spotrebičov v domácnosti, tak narastá aj počet diaľkových ovládačov na stole v obývacej miestnosti, počet vypínačov na stenách, a samozrejme, aj množstvo času na skontrolovanie stavu týchto spotrebičov. Preto vzniká dopyt po jednom jedinom zariadení, ktoré dokáže ovládať všetky, resp. väčšinu elektrospotrebičov v domácnosti.

1.1 Multimédia

Takýchto zariadení nájdeme na trhu veľmi veľa. Od úplne lacných až po drahšie a kvalitné, ktoré zaručujú kompatibilitu s konkrétnymi zariadeniami. Toto sa týka najmä tzv. čiernej techniky (televízna technika, audio a video prehrávače atď.). V ďalšom kroku sa prešlo od centralizovania ovládačov k centralizovaniu konkrétnych zariadení. Na trh prichádzajú tzv. set-top boxy. Set-top box je zariadenie, ktorého hlavnou úlohou je príjem pozemného digitálneho signálu. Ak však do set-top boxu privedieme vstupný signál z optickj linky od internetového poskytovateľa, tak set-top box spraví z nášho televízneho prijímača multimediálne centrum. Naš televízor môžeme teda okrem pozerania televíznych kanálov používať na surfovanie po internete, kontrolu e-mailov, sťahovanie filmov zo vzdialených digitálnych videopožičovní, a pod.

1.2 Kúrenie

Patrí k prvým systémom, ktoré sa v domácnostiach začali centralizovať. Prešlo vývojom od jednotlivých topných jednotiek v každej miestnosti, po centrálnu topnú jednotku umiestnenú niekde v kotolni a distribujúcu teplo pomocou ohriatej vody do všetkých častí domu. Od centrálnej riadiacej jednotky očakávame, že nám zariadi požadovanú teplotu v závislosti od toho, či je deň alebo noc, resp. či sme doma alebo v práci, a pod. Preto tieto jednotky vyžadujú väčšie množstvo snímačov teploty a schopnosť ovládať jednotlivé topné jednotky v izbách. K tomu je potrebná

ďalšia kabeláž. Preto je výhodné tento systém zakomponovať do jedného veľkého centralizovaného systému. Aj takýto systém využívajú dnešné „inteligentné domy“.

1.3 Inteligentné domy

Inteligentný dom je dom, v ktorom dokážeme ovládať takmer všetko z jedného miesta. A to aj zo vzdialeného miesta prostredníctvom internetu. Centralizuje nie len spomínané systémy, ale aj ďalšie iné, do jedného rozsiahleho systému so spoločným riadením. Ponuka riešenia inteligentných domov okrem iných obsahuje aj tieto časti:

- riadenie osvetlenia domácností (intenzitu osvetlenia, režimy osvetlenia pre jednotlivé miestnosti, automatické osvetľovanie, snímanie osvetlenia zvonku),
- pripájanie a odpájanie zásuviek už v rozvodnej skrini,
- zabezpečovací systém, kamery, simulácia prítomnosti osôb počas dovolenky,
- riadenie teploty v jednotlivých izbách,
- ozvučenie miestností (rôzna hudba v každej izbe podľa požiadavok užívateľa),
- meteorologická stanica,
- prehľad spotreby energie,
- ovládanie televízie a prístup na internet,
- domáce roboty,
- zavlažovanie trávniku,
- ovládanie týchto funkcií pomocou riadiacej jednotky s dotykovým displejom a pomocou diaľkového ovládača alebo internetu.

Implementovanie týchto systémov do domu sa realizuje už vo fáze projektovania. Podľa vopred pripraveného návrhu sa implementuje kabeláž, ktorej je pri týchto domoch pomerne veľa, keďže hovoríme o zbernicovej technológii, tak ku každému zariadeniu, ktoré chceme ovládať, musí viesť zbernicový kábel. K zariadeniam pracujúcim s napätím väčším ako 12-15 V musia viesť, samozrejme, aj klasické napájacie káble zo siete. Pri napätí do 15 V sme schopní tieto zariadenia napájať zo zbernice.

1.3.1 Káblové systémy

Ak chceme, aby zariadenia komunikovali, musia byť navzájom prepojené. Ak si však predstavíme, že ku každému zariadeniu, ktoré chceme ovládať, má viesť kábel, tak je to dosť odstrašujúca predstava. A aj napriek tomu sa to využíva. Spočíva to však v tom, že sa takýto systém musí do domu implementovať už pri jeho stavbe, aby tie káble boli skryté. Ale aj keď sú skryté, tak tam sú, a niekto ich musí zaplatiť. Preto je toto riešenie finančne náročné. A do už postaveného domu prakticky nezrealizovateľné bez stavebných zásahov.

EIB(European Installation Bus)

Je to zbernicová technológia, ktorá je určená na riadenie zariadení v domoch. Je napájaná napätím 24V. Zariadenia, ktoré sú na ňu pripojené, navzájom komunikujú v režime *peer to peer*. Žiadne zariadenie nemá vyššie postavenie. Každé zariadenie má svoju adresu a integrovaný mikroprocesor, na základe ktorého sa rozhoduje. Takýto systém riadenia je decentralizovaný. Zariadenia medzi sebou komunikujú pomocou datagramov, ktoré si zasielajú. Na zbernici sa dajú vytvoriť aj určité podmienené väzby. Napríklad, ak je otvorené okno, tak sa nezapne klimatizácia.

CIB(Common Installation Bus)

Jedná sa taktiež o zbernicovú technológiu. Je podobná ako EIB, ale zariadenia na nej komunikujú v režime master-slave. To znamená, že zariadenia v režime slave nemajú právo vysielat', kým neprijmu požiadavku zo stanice v režime master. Zbernica CIB má veľký dosah a je ľahko rozširiteľná. Na jednu vetvu centrálnej jednotky je možné pripojiť až 32 zariadení. V prípade potreby sa dajú k centrálnej jednotke pripojiť ďalšie master moduly, čo umožní pripojenie ďalších 32 zariadení pre každý master modul. Každé zariadenie na CIB zbernicu má svoju unikátnu 16 bitovú adresu. Táto adresa je uvedená na kryte každej jednotky. A programátor má možnosť zistiť si ju aj elektronicky. Systém obsahuje dva 12 voltové akumulátory, ktoré v prípade výpadku energie udržia celý systém v prevádzke.

RS485

RS485[12] je dvojvodičová zbernicová technológia. Vodiče sú oznčené A a B alebo „+“ a „-“. Maximálna dĺžka zbernice je 1200 m. Maximálna prenosová rýchlosť

je 10 Mb/s. Konce vodičov musia byť z každej strany zakončené odpormi $110\ \Omega$. Logické stavy sú reprezentované rozdielným napätím medzi vodičmi. Ako vodič je používaná krútená dvojlinka (twisted pair). Komunikácia na RS485 je tiež master-slave. Zbernica je napájana napätím 15V. Maximálny počet uzlov je 32. Sú číslovane 0-31.

Zaujímá nás však aj iná technológia, ktorá je v spolupráci so ZigBee ideálnym riešením pre už postavené domy. Je to technológia PLC (Power Line Communication), čiže technológia komunikácie po silových rozvodoch. Táto technológia výrazne ušetrí finančné prostriedky na infraštruktúru.

PLC

Systémy PLC sú rozdelené do dvoch oblastí. Úzkopásmové a širokopásmové. Úzkopásmové systémy sa používajú pre automatizáciu a riadenie a širokopásmové najmä pre prístup k internetu a realizáciu lokálnych sietí LAN.

Hlavným nedostatkom tejto technológie je, že na silových rozvodoch dochádza k rušeniu z rôznych spotrebičov, ktoré sa dopredu nedá odhadnúť. Preto je potrebné zvoliť vhodnú moduláciu. Najčastejšie používané modulácie sú **OFDM**(Orthogonal frequency-division multiplexing) a **DCSK**(Differential Code Shift Keying).

1.3.2 Bezdrôtové systémy

Ďalším riešením sú bezdrôtové technológie. Tu sa nám ponúka viac možností. Na diaľkové ovládanie spotrebičov sa momentálne používa najčastejšie IrDA. To vyžaduje optický kontakt, čo je nepraktické. V domácnostiach sa používa aj technológia WiFi. Je to technológia pre prenos dát siete internet. Je to však pomerne drahá technológia a implementovať ju do každého zariadenia, ktoré chceme ovládať je nemožné. Najvýhodnejšou bezdrôtovou technológiou pre naše využitie je technológia ZigBee (ZigBee Alliance)[13]. Je to hardverovo a energeticky nenáročná technológia. Neposkytuje takú rýchlosť prenosu ako technológia WiFi, ale pre kontrolu a riadenie je ideálna.

1.3.3 Najpoužívanéjšie systémy inteligentných domov

Systém LonMark

Systém Lon[9] je ďalším otvoreným systémom, ktorý k nám prišiel z USA. Tiež obsahuje veľké množstvo rôznych zariadení a umožňuje komunikáciu po krútenej dvojlinke, po vedení 230 V, ale aj bezdrôtový prenos. Riešenia LON sa implementujú aj na najvyššiu úroveň riadenia zariadení a do systémov merania a regulácie.

Systém INELS

Základom systému INELS[4] je centrálna jednotka CU2-01M, ktorá obsahuje rozhranie Ethernet. Používa zbernicu CIB(Common Installation Bus). Zbernica je napájaná napätím 27 V. Je na nu možné pripojiť GSM modem, čo umožní ovládanie zariadení pomocou GSM telefónu. INELS má k dispozícii tiež veľké množstvo modulov; či už snímacích, riadiacich alebo zobrazovacích.

Systém KNX

KNX[8] je otvorený systém, ktorý nadviazal na EIB (European Installation Bus) nazývaný tiež Instantbus a pričlenil ďalšie dva systémy EHS (Electronic Home System) a BatiBus. Princíp tejto technológie je založený na vzájomnej komunikácii jednotlivých prístrojov prepojených dátovým káblom. Štruktúra siete je rozložená na tri úrovne. Najvyššia je centrálna úroveň, ktorá obsahuje 15 tzv. línii strednej úrovne a každá z týchto línii obsahuje 15 ďalších línii spodnej úrovne. Každá línia ma vlastné napájanie a je na nu možné napojiť až 64 zariadení. Zariadenia sú rozdelené na snímače, ovládacie prvky a akčné členy. Každý prvok sa dokáže rozhodovať sám. Tým však narastá jeho cena.

Systém BatiBus sa zameriava na oblasť vykurovania, klimatizácií a vetrania. Systém EHS ponúka najmä rozhranie k bielej technike (chladničky, umývačky riadu, sporáky a pod.). Spojením týchto systémov prináša veľa možností pri nastavovaní optimálneho prostredia a kontrolu stavu spotrebičov.

2 REALIZÁCIA RIADIACEHO SYSTÉMU

Našou úlohou je vybrať vhodnú riadiacu jednotku, implementovať do nej operačný systém, navrhnúť pre ňu aplikáciu s grafickým rozhraním, ktoré bude priateľské pre užívateľa a využiteľné v domácnosti. A aby bola schopná komunikovať s ďalšou aplikáciou, ktorej úlohou bude počúvanie na sériovom porte, a prijaté hodnoty zapisovať do tabuľky. Tieto hodnoty budú vyhodnocované a znázornené grafom. Neskôr bude táto aplikácia slúžiť aj na ovládanie určitých elektrických zariadení v dome.

2.1 Výber riadiacej jednotky

Terajšie riešenia inteligentných domov majú riadiacu jednotku umiestnenú na DIM lište v rozvážači a všetky ostatné periférne jednotky s ňou komunikujú. Resp. v decentralizovanom systéme má každá jedna periférna jednotka svoj mikroprocesor, na základe ktorého sa rozhoduje a je schopná komunikovať s ostatnými zariadeniami samostatne. My sme sa rozhodli pre riadiacu jednotku skombinovanú s užívateľským grafickým rozhraním, ktorá bude umiestnená v obytných priestoroch. Bude ovládaná dotykovou obrazovkou a bude mať prístup do siete.

2.1.1 SBC(Single-Board Computer)

Je to kompletný počítač ako ho poznáme, ale je umiestnený na jednej doske. Obsahuje všetky prvky ako plne funkčný počítač, a taktiež všetky vstupno-výstupné porty, aké poznáme. Je viacero výrobcov, ktorý ponúkajú tieto *embedded* (vstavané) systémy. Systémy od rôznych výrobcov sa málo líšia výkonom a cenovo sú veľmi rozdielne. Niektorí výrobcovia využívajú procesory Intel a AMD, ktoré rozhodne patria k tým drahším. Preto sme sa zamerali na procesory ARM. Procesory ARM sú úspornejšie ako ostatné procesory, a ich cena je priateľná vzhľadom na ich výkon.

2.1.2 Procesory ARM

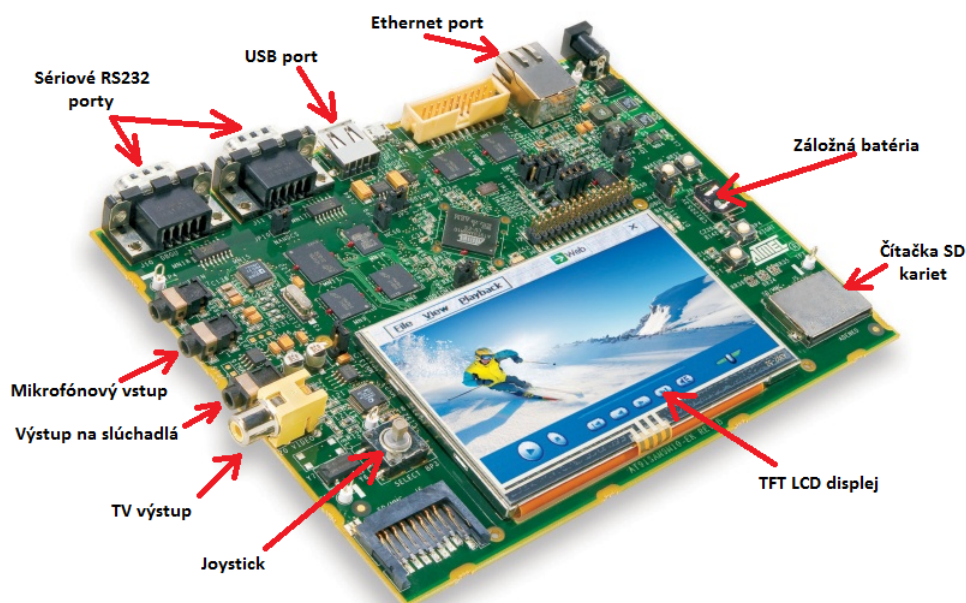
Procesor si v dnešnej dobe predstavujeme ako malý, integrovaný obvod, resp. čip. Niekedy to tak však nebolo a procesory boli obrovské skrine plné tranzistorov. Dokonca až celé poschodia plné takýchto skrií. Po príchode integrovaných obvodov sa začali rapídne zmenšovať a začal narastať ich výkon. Postupne sa zvyšovala frekvencia, s akou procesor dokázal spracovávať inštrukcie, a taktiež sa zväčšovala dĺžka slova, s ktorým dokázal procesor pracovať. Frekvencie sa vyšplhali až na niekoľko gigahertzov a dĺžka slova na 64 bitov. Ďalšie zvyšovanie týchto parametrov prináša viacero problémov, preto sa prešlo na viacjadrové procesory. V jednom čipe môže byť odteraz viac jadier, ktoré dokážu spolupracovať a prácu si podeliť. Takéto procesory sú už však drahšie a ich výkon si vyžaduje aj efektívne chladenie. Takéto chladenie však zaberá oveľa viac priestoru ako samostatný procesor. Takže čím väčší výkon, tým väčší chladič.

Niektoré zariadenia však nepotrebujú obrovský výkon procesoru. Majú ale iné požiadavky. Napríklad nízka spotreba, a čo najmenšie rozmery. Čo automaticky vylučuje veľký chladič. Preto sa vývoj mikroprocesorov posúva aj týmto smerom. A na trh prichádzajú procesory s dostatočným výkonom, avšak malou spotrebou. Dobrým príkladom sú momentálne malé notebooky tzv. „netbooky“ využívajúce hlavne mikroprocesor Intel Atom. Intel je špičkou na trhu mikroprocesorov pre osobné počítače a notebooky. Najväčším konkurentom Intel-u sa zdá byť AMD. V najbližšej dobe sa môže začať konkurenčný boj. Nie však medzi AMD a Intelom ale medzi Intelom a spoločnosťou ARM. ARM pozná málokto a pritom je na svete niekoľkonásobne viac mikroprocesorov od firmy ARM, než od firmy Intel. Práve preto, že sa ARM vybrala opačným smerom ako Intel. Smerom úsporných mikroprocesorov. V dnešnej dobe, každý jeden telefón obsahuje mikroprocesory firmy ARM. Keďže Intel sa snaží svojím výkonným procesorom znížiť spotrebu (Intel Atom) a ARM sa snaží svojím úsporným procesorom zvýšiť výkon (Cortex A9), môžeme čoskoro očakávať konkurenčný boj na trhu s netbookmi. Dobrým príkladom je Google. Ich nový netbook, resp. „smartbook“, ako ho nazývajú, obsahuje procesor ARM a operačný systém od Google. Dokonca aj spoločnosť Microsoft vyhlásila, že ich Windows 8 bude podporovať procesory ARM.[7]

2.1.3 Riadiace jednotky

SAM9M10

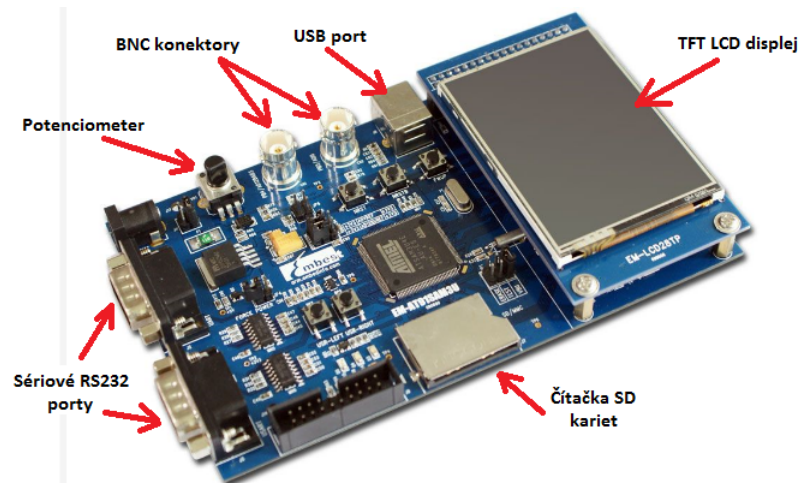
- výrobca firma ATMEL,
- 400 MHz procesor ARM9,
- 32 DMA kanálov,
- podporuje DDR2 pamäť,
- 100Mb/s Ethernet port,
- dotykový TFT LCD displej,
- joystick,
- cena 486 EUR.



Obr. 2.1: SBC SAM9M10

SAM3U

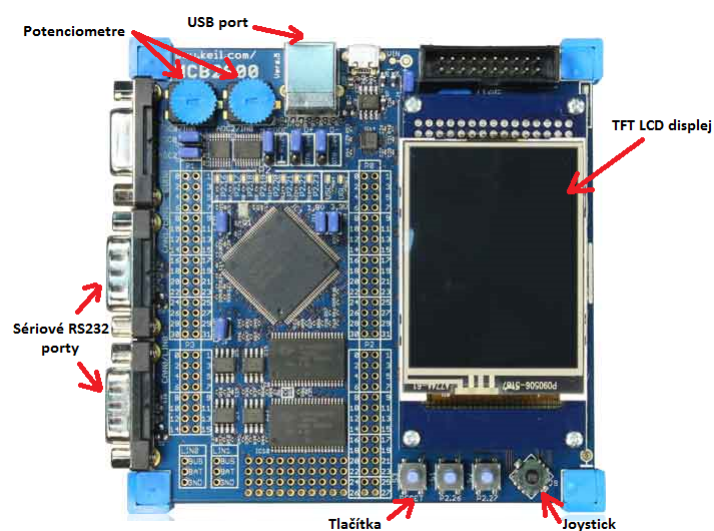
- výrobca firma ATMEL,
- 96MHz procesor ARM Cortex-M3,
- 22 DMA kanálov,
- dotykový TFT LCD displej,
- cena 212 EUR.



Obr. 2.2: SBC SAM3U

MCB2929

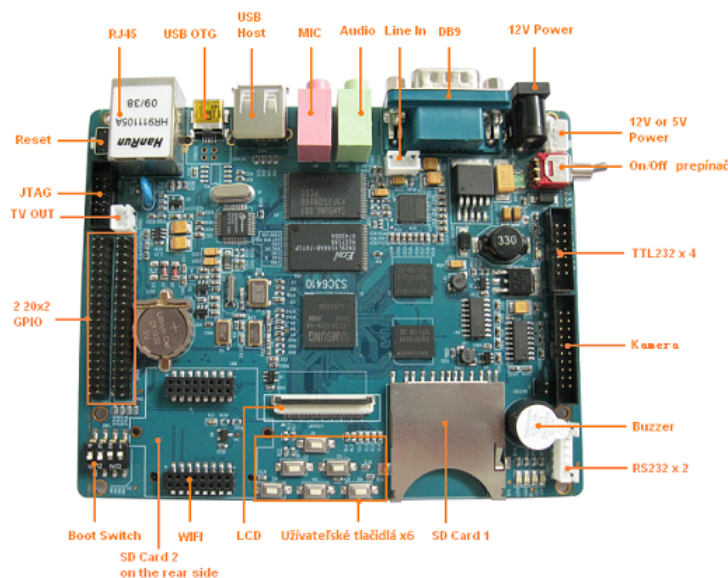
- výrobca firma KEIL,
- 125MHz procesor ARM9,
- joystick,
- dotykový TFT LCD displej,
- cena 242 EUR.



Obr. 2.3: SBC MCB2929

MCB2929

- výrobca firma Embedsky,
- 667MHz procesor ARM11,
- 100Mb/s Ethernet port,
- modul s webkamerou,
- podporuje WiFi modul,
- TV výstup,
- 6 užívateľských tlačidiel + tlačidlo reset,
- dotykový TFT LCD displej,
- cena 190 EUR.



Obr. 2.4: SBC6410

Vďaka nízkej cene a rýchlej dostupnosti sme sa rozhodli pre SBC6410. Obsahuje procesor ARM11, ktorý pracuje na frekvencii až 667MHz, čo nám poskytuje dostatočne vysoký výkon na prevádzkovanie užívateľskej grafickej aplikácie aj s prípadnými multimedálnymi prvkami. Bohužiaľ však nemá takú technickú podporu ako FriendlyArm, ktorý bol pôvodne v pláne. Priložený manuál je nepostačujúci. S týmto návodom však dokážeme inicializovať aj jednotku od FriendlyArm.

2.2 Inicializácia riadiacej jednotky

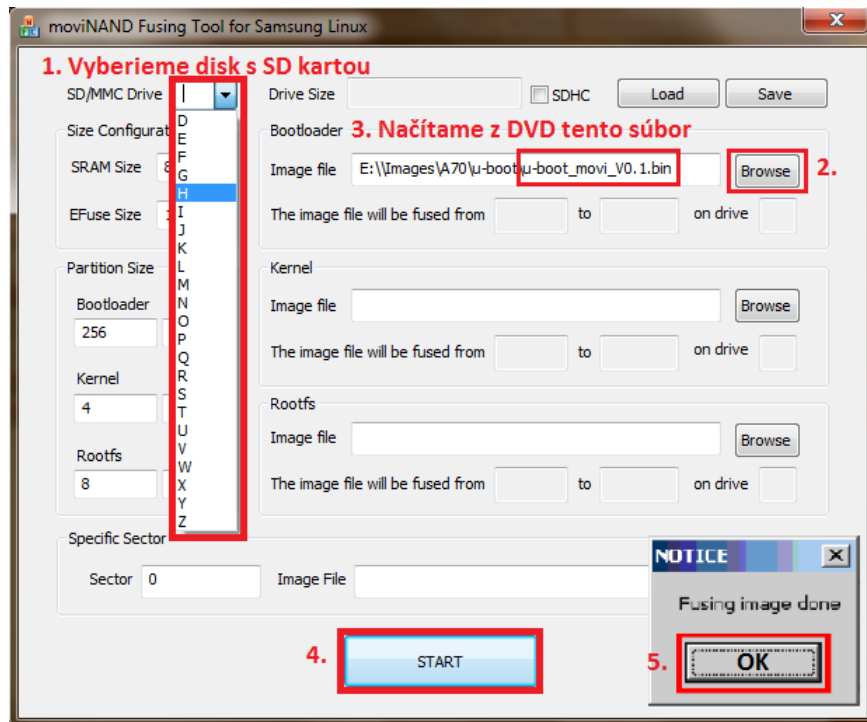
Balenie SBC6410 obsahuje tieto časti:

- hlavná doska s procesorom,
- 7 palcová LCD dotyková obrazovka,
- flex kábel k obkrazovke,
- UTP kábel na pripojenie do siete,
- RS232 kábel na sériové pripojenie k počítaču,
- USB kábel,
- modul s webkamerou,
- napájací zdroj,
- DVD s manuálmi a zdrojovými súbormi pre OS (Android, Linux, WindowsCE).

Rozhodli sme sa pre operačný systém Linux, pretože nám umožňuje zásahy do svojich zdrojových kódov a jednoduchší prístup k portom ako OS WindowsCE. Chceli sme, aby sa naša riadiaca jednotka kompletne bootovala zo siete, aby sme šetrili jej integrovanú flash pamäť. To sa nám však s touto jednotkou nepodarilo. Teraz funguje jednotka tak, že po spustení si zo siete nabootuje len jeden priečinok, v ktorom je naša spustiteľná aplikácia.

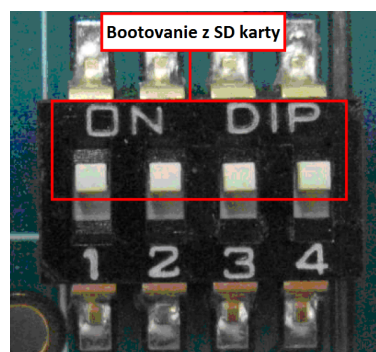
2.2.1 Postup inicializácie

1. Najprv potrebujeme na SD kartu vypáliť image súbor s bootovacím programom U-Boot. Tento image súbor vypálime pomocou programu „moviNAND_Fusing_Tool.exe“, ktorý nájdeme na priloženom DVD v priečinku Tools. Musíme si dať pozor, aby bola SD karta naformátovaná v tabuľke FAT (FAT16). SD karta naformátovaná v inej tabuľke nemôže byť použitá na bootovanie.



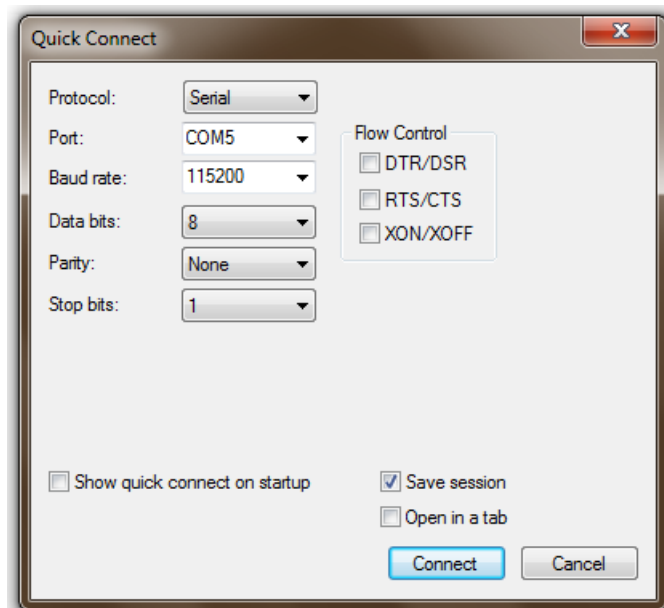
Obr. 2.5: Vytváranie bootovacej SD karty

2. Kartu vložíme do našej riadiacej jednotky a nastavíme bootovacie prepínače na doske do polohy na bootovanie z SD karty.



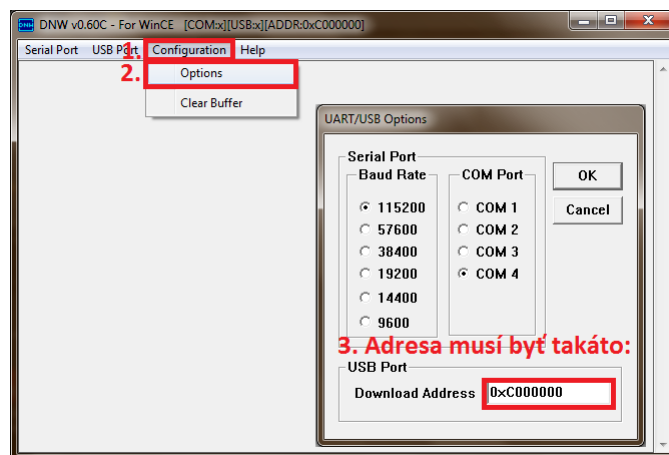
Obr. 2.6: Nastavenie prepínačov do polohy pre bootovanie z SD karty

3. Potrebujeme mať na počítači nejaký terminál, ktorý nám umožní komunikáciu po sériovej linke. Pre OS Windows je to napríklad program Secure CRT, ktorý nájdeme v priečinku Tools na priloženom DVD. Pre OS Linux sme používali terminál Minicom.



Obr. 2.7: Nastavenie pripojenia v Secure CRT

4. Musíme ešte nastaviť program DNW. Ten slúži na zasielanie image súborov do našej riadiacej jednotky cez USB kábel.

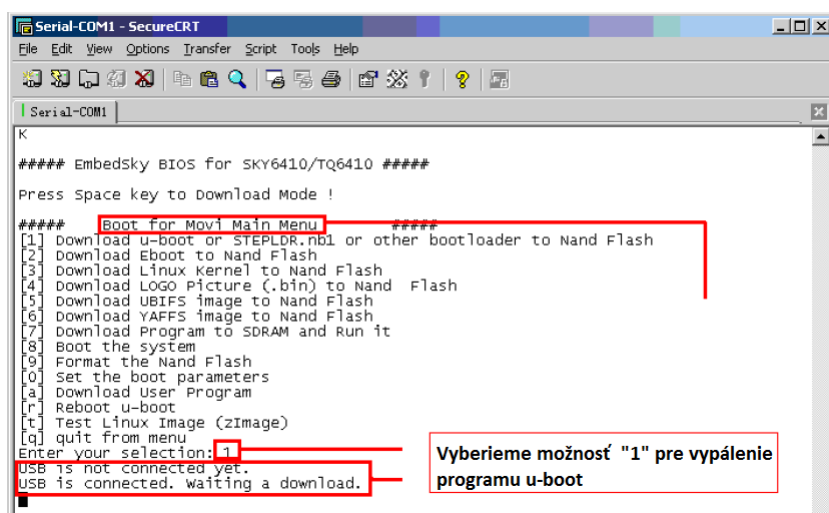


Obr. 2.8: Nastavenie USB pripojenia v DNW

5. Teraz sa už môžeme pustiť do inštalácie Linuxu do našej riadiacej jednotky. Riadiacu jednotku prepojíme s našim počítačom pomocou RS232 káblu a spustíme program Secure CRT. Ďalej tieto dve zariadenia prepojíme pomocou USB káblu a spustíme program DNW. Vložíme do riadiacej jednotky SD kartu s na-

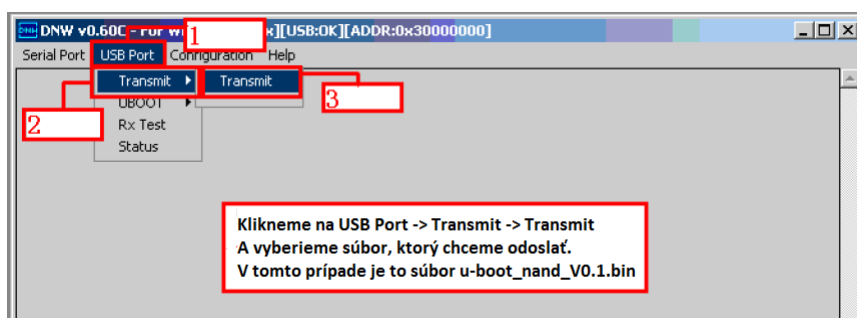
páleným U-Boot a nastavíme bootovacie prepínače ako bolo na obr.2.6. Pripojíme napájanie jednotky a na boku našej jednotky ju spustíme prepnutím prepínača ON/ OFF.

6. Ak sme správne nastavili program Secure CRT, malo by sa nám tam ukázať základné U-Boot menu. Pomocou neho vypálime potrebné image súbory do flash pamäte našej riadiacej jednotky. Najprv do nej potrebujeme vypáliť U-Boot.



Obr. 2.9: Vypálenie U-Boot do NAND flash

Program U-Boot teraz čaká na image súbor na USB linke. Ten mu odošleme pomocou programu DNW. Názov súboru je „u-boot_nand_V0.1“.

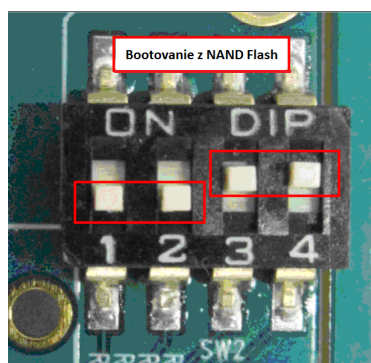


Obr. 2.10: Odoslanie image súboru pomocou DNW

7. Týmto spôsobom vypálime aj ďalšie dva súbory. Najprv Linux kernel. V U-Boot menu tentokrát zvolíme možnosť 3 a cez DNW odošleme súbor s názvom

„zImage_A70_V0.1“. Posledným suborom je filesystem Linuxu. V U-Boot menu možnosť 6 a cez DNW odošleme súbor „root_qtopia_2.2.0_6410_V0.1“.

8. Teraz už máme nainštalovaný Linux na našej riadiacej jednotke. Vypneme ju, vyberieme SD kartu a bootovacie prepínače dáme zasa do polohy pre bootovanie z NAND Flash.



Obr. 2.11: Nastavenie prepínačov do polohy pre bootovanie z NAND Flash

9. Po opätovnom spustení jednotky sa nabootuje systém Linux z NAND Flash pamäte. Je to distribúcia Qtopia 2.2.0. Obsahuje základné programové vybavenie, pár hier a ukážkové médiá ako video, hudba a obrázky. Odporúčame však použiť distribúciu QT 4.7. Pri kompilovaní Qtopie 2.2.0 sme mali rôzne problémy a nakoniec sa nám crosskompiláciu pre ARM ani sfunkčnilo. Návrh grafickej aplikácie je teda v prostredí QT 4.7.

2.2.2 Bootovanie zo siete

NFS (Network File System) je systém, ktorý nám umožňuje prístup k súborom na inom počítači (NFS Server) a používať ich ako naše vlastné. Týmto šetríme pamäť na našom disku a pracovné využitie nášho disku. Pôvodne sme chceli, aby sa zo serveru bootoval celý Linux file system aj s kernelom. To nám však s našou periférnou jednotkou nefungovalo. Tak sme to bootovanie obmedzili na pracovný priečinok, v ktorom budú naše testovacie programy. Popíšeme si postup, ako nastaviť NFS:

1. Na periférnej jednotke si nastavíme pevnú IP adresu. Musí byť z rozsahu siete, ktorú používame.
2. Ako pracovnú stanicu používame počítač s OS Linux (distribúcia Ubuntu 10.10). Na túto pracovnú stanicu si z DVD priloženého k nášmu kitu, roztarujeme všetky súbory z priečinku Linux. Tieto súbory budú v našom domovskom priečinku s cestou „/opt/EmbedSky/“ v priečinku EmbedSky si vytvoríme náš vlastný priečinok. Pomenovaný ako „rootfs“. Tento priečinok budeme bootovať pre našu perifernú jednotku.
3. Nainštalujeme si NFS server na našu pracovnú stanicu. To spravíme príkazom:

```
sudo apt-get install nfs-kernel-server
```

4. Nastavíme NFS server. Do súboru „/etc/exports“ pridáme priečinky, ktoré chceme, aby boli zdieľané. V našom prípade to priečinok „/opt/EmbedSky/rootfs“. Do súboru „/etc/exports“ preto pridáme tento riadok:

```
/opt/EmbedSky/rootfs 192.168.1.6/255.255.255.0  
(rw,sync,no_root_squash,no_subtree_check)
```

Kde IP adresa je IP adresa našej periférnej jednotky.

5. Na kite vytvoríme priečinok „/mnt/test“ a príkazom :

```
mount -o nolock,vers=2 192.168.1.1:/opt/EmbedSky/rootfs /mnt/test
```

načítame tento priečinok zo serveru do nášho kitu. IP adresa je adresa našej pracovnej stanice (NFS Server).

6. Konfiguráciu otestujeme vytvorením jednoduchej aplikácie Hello World vytvorenej na našej pracovnej stanici a skompilovanej do nášho zdieľaného priečinka „/opt/EmbedSky/rootfs“. Kód aplikácie je :

```
#include <stdio.h>  
  
int main()  
{  
    printf("Hello world!");  
    return 0;  
}
```

Textový súbor uložíme s názvom „hello.c“ a príponou „.c“. Skompilujeme ho pomocou kroskompilátoru použitím príkazu :

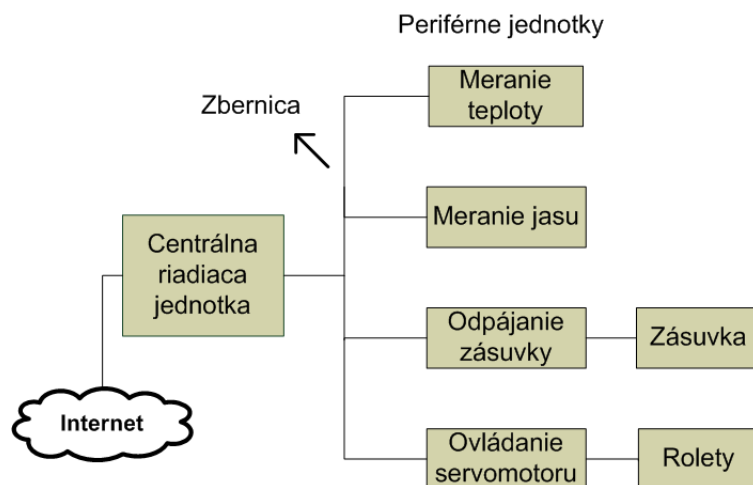
```
arm-linux-gcc hello.c -o hello
```

Tieto súbory sa nám zobrazia aj na našej riadiacej jednotke. Našu aplikáciu spustíme na kite príkazom:

```
./hello
```

2.3 Návrh systému

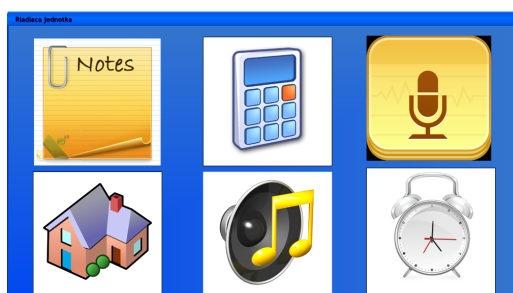
Chceme navrhnuť systém, v ktorom bude centrálna riadiaca jednotka zároveň jednotkou s grafickým užívateľským rozhraním. Táto jednotka bude po zbernici komunikovať s periférnymi jednotkami pomocou jednoduchého komunikačného protokolu. Periférne jednotky budú merať napätie, teplotu, jas, atď. A taktiež budú schopné odpájať a pripájať elektrické obvody, či už svetelné alebo napájacie. Popríklad budú schopné ovládať servo motory na zaťahovanie roliet, resp. garážových dverí. Naša centrálna jednotka bude pripojená na internetovú sieť a bude na nej fungovať webový server. To umožní kontrolu domácnosti na diaľku, resp. jej ovládanie.



Obr. 2.12: Blokova schéma systému

2.3.1 Návrh aplikácie pre riadiacu jednotku

Naša riadiaca jednotka má 7" dotykový displej a mala by byť zabudovaná v stene. Budeme preto navrhovať aplikáciu, ktorú sme schopní ovládať dotykmi na displeji. To znamená, čo najväčšie tlačítka. Aplikácia bude v obytných priestoroch, preto by mala byť čo najviac využiteľná. Integrujeme do nej okrem ovládania domácnosti aj možnosť zápisu poznámok, prípadne hlasových záznamov, jednoduchú kalkulačku, budík a prehrávač hudby. Pri nečinnosti sa obrazovka buď vypne, alebo sa spustí šetrič obrazovky vo forme analógových hodín.



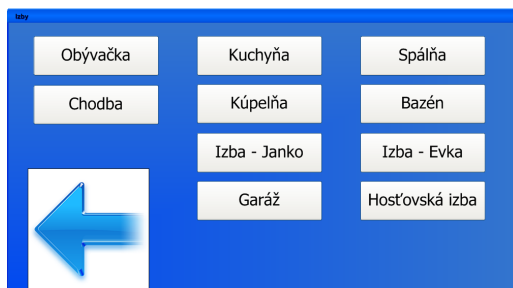
Obr. 2.13: Hlavné menu aplikácie

Ikonka domu symbolizuje ovládací režim, v ktorom je možné ovládať jednotlivé miestnosti, resp. celý dom. Po kliknutí na túto ikonku sa nám otvorí ďalšie menu, v ktorom máme možnosť ovládať osvetlenie, teplotu a zatiahnute roliet pre celý dom. Máme tu ešte tlačítko miestnosti. Po kliknutí na toto tlačítko sa nám otvorí ďalšie menu so zoznamom miestností v dome. A ešte je tu tlačítko so šípkou na návrat do predchádzajúceho menu.



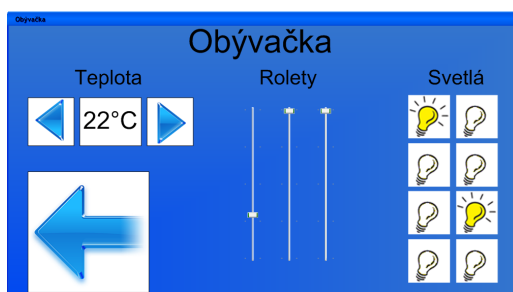
Obr. 2.14: Ovládacie menu domu

Po kliknutí na ikonku izby sa nám zobrazí ďalšie menu, ktoré bude vyzeráť približne takto:



Obr. 2.15: Menu výberu izby

Vyberieme si zo zoznamu izbu a zobrazí sa nám menu pre konkrétnu izbu. Pre každú izbu bude iné; podľa toho, koľko svetiel a roliet je v izbe. Zobrazí sa nám aktuálna teplota v izbe a šípkami si ju môžeme upraviť na požadovanú teplotu. Ďalej tam máme posuvníky na nastavovanie roliet. Podľa toho, v akej výške je posuvník, v takej výške bude aj roleta. Môžeme nastavovať aj svetlá. Po kliknutí na jednotlivé žiarovky v menu sa budú zapínať alebo vypínať svetlá, resp. svetelné scény v izbe.



Obr. 2.16: Ovládanie konkrétnej izby

Každé zariadenie v dome má definovanú svoju vlastnú premennú. Táto premenná sa po štarte aplikácie načíta z externého súboru. Tomuto súboru sme dali XML formu. Kvôli prehľadnosti a priamejšiemu prístupu k dátam ako keby to bol len textový súbor. Štruktúru nášho XML súboru môžeme vidieť na obr.2.17.

```

<?xml version="1.0" encoding="UTF-8"?>
- <dom>
  + <obyvacka>
  - <kuchyna>
    + <svetlo id="1">
    - <svetlo id="2">
      <hodnota>0</hodnota>
    </svetlo>
    + <svetlo id="3">
    + <svetlo id="4">
    - <roleta id="1">
      <hodnota>40</hodnota>
    </roleta>
    + <roleta id="2">
    + <roleta id="3">
  </kuchyna>
</dom>

```

Obr. 2.17: Štruktúra XML súboru.

2.4 Praktická realizácia

2.4.1 Realizácia aplikácie pre riadiacu jednotku

Ako som už spomínal, nepodarilo sa nám skompilovať Qtopiu 2.2.0 a preto je aplikácia realizovaná v prostredí QT 4.7 od Nokie.[10] Najvhodnejšie je stiahnuť si celý SDK balík priamo na stránke Nokie. [11] Tento balík si potom môžeme skompilovať buď pre klasickú architektúru, náš linux na počítači, alebo pre architektúru embedded systému. V našom prípade je embedded systém založený na procesore ARM. Ak chceme kompilovať pre ARM, tak parametre pri procese configure musia byť takéto:

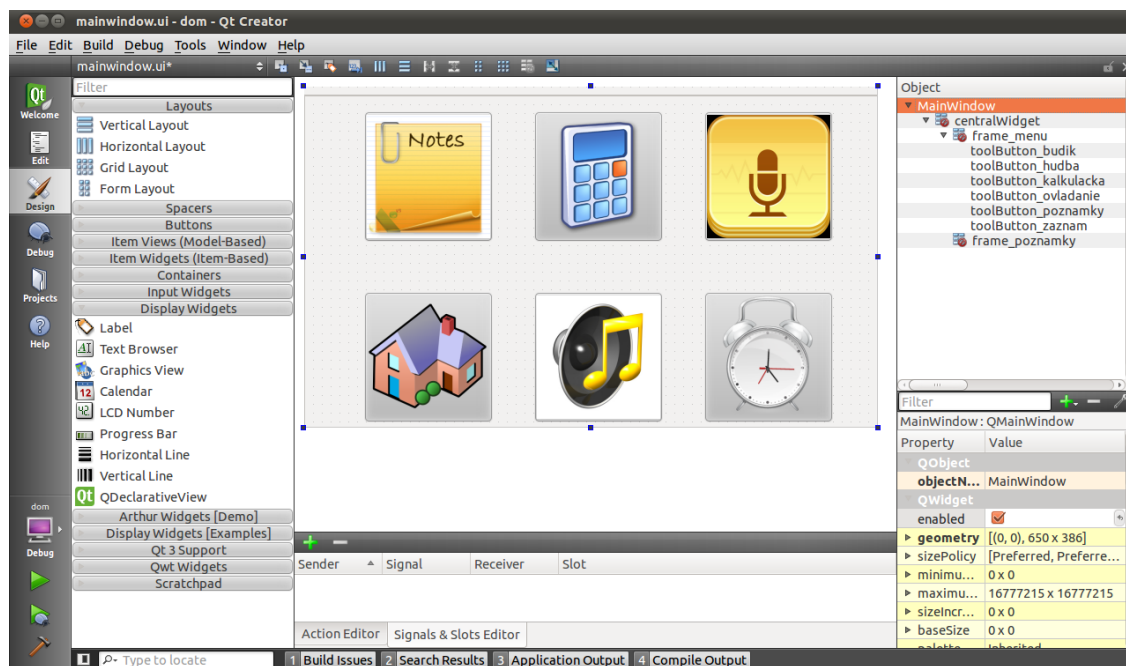
```

./configure --opensource --embedded arm --xplatform qws/linux-arm-g++
--no-webkit --qt-libtiff --qt-libmng --qt-mouse-tslib --qt-mouse-pc
--no-mouse-linuxtp

```

My však na našom SBC6410 nemáme QT 4.7, preto pri kompilácii použijeme „configure“ bez parametrov aby sme mohli aplikáciu testovať na našom linuxe. Po nainštalovaní si spustíme QT Creator. Je grafický nástroj na vytváranie grafických rozhraní a rôznych „layout“-ov. Má niekoľko režimov. My budeme využívať grafický režim pre editovanie súboru s príponou .ui a textový režim pre editáciu zdrojových súborov .cpp a hlavičkových súborov .h.

V grafickom režime si navrhujeme ako bude naša aplikácia vyzerieť. Návrh sme si popísali už skôr v kapitole 2.3.1, takže nám zostáva už len realizácia a naklikáť si jednotlivé tlačítka. Pri realizácii využívame objekty s názvom „frame“. Na tieto objekty si naukladáme tlačítka do požadovaného tvaru. Po kliknutí na tlačítko využijeme metódu, pri ktorej sa nám jeden frame skryje a nový sa zobrazí. Nový frame obsahuje nové tlačítka a tlačítko na návrat na starý frame.



Obr. 2.18: Prvý frame v QT Creatore.

Takýmto spôsobom si vytvoríme aj ostatné frame-y. Priebežne si vo Void MainWindow nastavujeme, ktoré frame-i sa po spustení aplikácie skryjú.

```
ui->frame_ovladanie->hide();
```

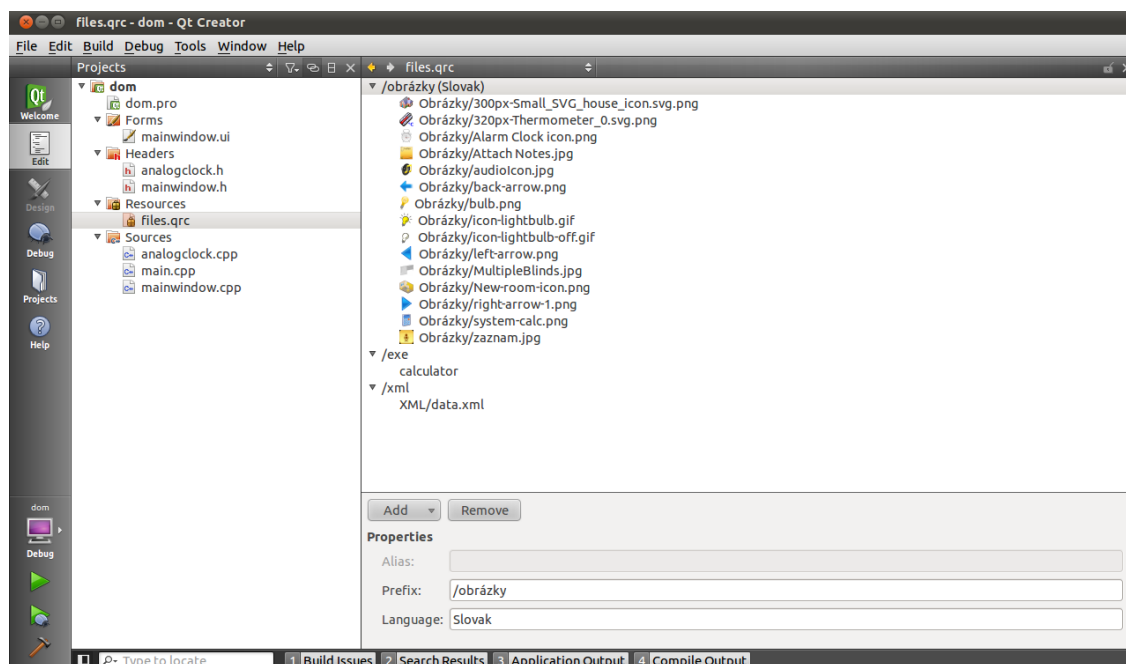
Nastaviť funkciu tlačítka môžeme spraviť ako klasický event v Céčku a deklarovaním si jeho funkcie. QT Creator to však dokáže spraviť za nás a to kliknutím pravým tlačítkom myši na tlačítko a vybrať možnosť „Go to SLOT“. Zobrazí sa nám menu, v ktorom si máme možnosť vybrať po akej akcii s tlačítkom sa má funkcia vykonať. V našom prípade je to akcia „clicked()“. Nesmieme zabúdať na vhodné pomenovanie objektov, aby sme sa v tom dokázali neskôr zorientovať.

```

void MainWindow::on_toolButton_ovladanie_clicked()
{
    ui->toolButton_poznamky->hide();
    ui->toolButton_kalkulacka->hide();
    ui->toolButton_zaznam->hide();
    ui->toolButton_ovladanie->hide();
    ui->toolButton_hudba->hide();
    ui->toolButton_budik->hide();
    ui->frame_ovladanie->show();
}

```

Jednotlivým tlačítkam nastavíme ikonky. Tieto ikonky však chceme zahrnúť do kompilácie a preto si vytvoríme nový súbor typu „resources“, ktorý má príponu .qrc, a pridáme ho do nášho projektu. Vytvoríme si v ňom prefix „obrázky“ a zadáme cestu k ikonkam. Výsledok by mal vyzeráť podobne ako na obr.2.19.



Obr. 2.19: Resources v QT Creatore.

V grafickom režime si potom cez vlastnosti tlačítka nájdeme nastavenie ikonky a ako zdroj vyberieme „resources“. Táto metóda funguje. Rovnako by to malo fungovať aj po napísaní tejto cesty cez príkaz:


```
ui->toolButton_obyvacka->setIcon( QIcon( ": / obrázky / Obrázky / nieco . png" ) );
```

Nám však tento spôsob z neznámeho dôvodu nefunguje. Takže aj prídavné spustiteľné súbory musíme adresovať len relatívnou cestou a nakopírovať si ich do zložky programu. Spustenie externého súboru realizujeme pomocou triedy „QProcess“. Vytvoríme si preto funkciu void kalkulacka().

```
void MainWindow::kalkulacka()
{
    proc = new QProcess( this ); // alokácia pamäte
    QString cesta="exe/calculator";
    proc->start(cesta); // štart programu
}
```

Po kliknutí na tlačítko s ikonou kalkulačky túto funkciu následne voláme.

```
void MainWindow::on_toolButton_kalkulacka_clicked()
{
    kalkulacka();
}
```

Nesmieme však zabúdať si všetky funkcie, ktoré si vytvárame sami v textovom režime, deklarovať v hlavičkovom súbore „mainwindow.h“.

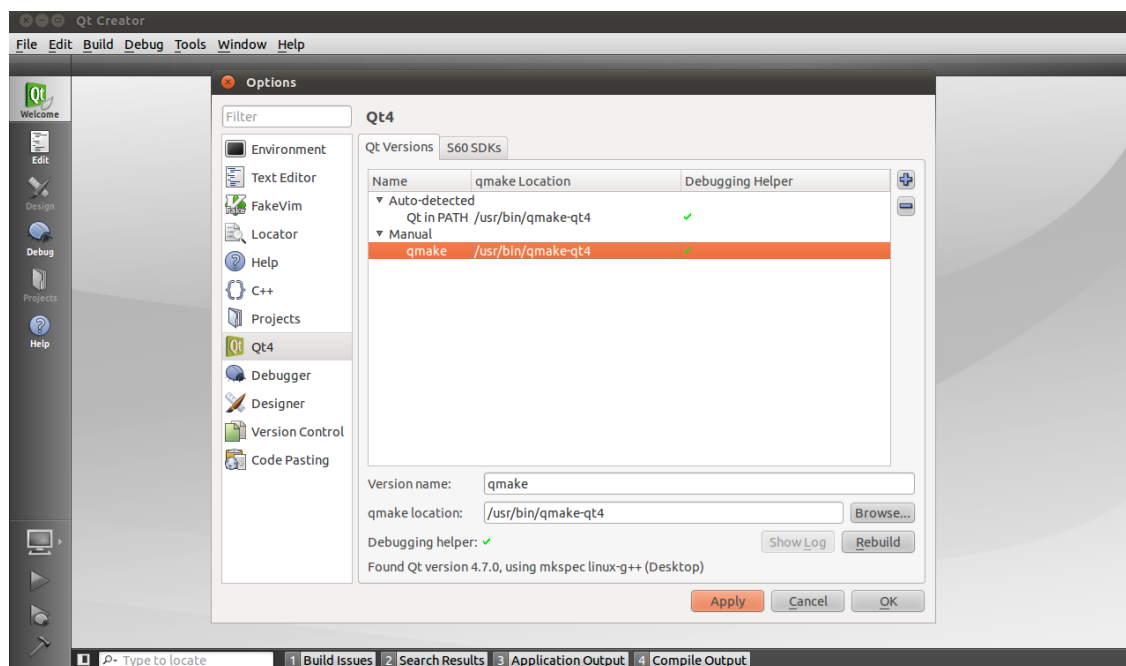
```
class MainWindow : public QMainWindow
{
    Q_OBJECT
public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();
    void kalkulacka(); // <-- deklarácia našej funkcie
private:
    Ui::MainWindow *ui;
private slots:
    void on_pushButton_spalna_svetlo_1_1_clicked();
    void on_verticalSlider_obyvacka3_valueChanged(int value);
    // . všetky ostatné sloty
    void on_toolButton_obyvacka_clicked();
};
```

Takýmto spôsobom dokážeme spraviť celú aplikáciu. Chceme však, aby si aplikácia po spustení načítala aktuálny stav zariadení, ktorý je zaznamenaný v XML súbore. Štruktúra tohto súboru je na obr.2.17. Na načítavanie dát z XML existuje niekoľko prístupov. Najznámejšie sú DOM (Document Object Model) a SAX (Simple API for XML). DOM spočíva v tom, že sa do pamäte načíta celý dokument a pracuje sa dátami objektovo. SAX funguje na princípe vyhľadávania textového reťazca v XML súbore a až potom zobrazí dáta. V našom prípade nie je XML súbor natoľko rozsiahly, aby sme ho nemohli používať v celku. A metóda objektového prístupu je pre nás výhodná. Preto využijeme DOM. Základný princíp je v tom, že si načítame koreňový element do premennej a postupne si do premenných uložíme aj jeho uzly, ktoré triedime podľa názvu.

```
QFile* file = new QFile("XML/data.xml"); //načítanie nášho súboru
//Ak súbor nedokážeme otvoriť, ukáže sa chybová správa.
if (!file->open(QIODevice::ReadWrite | QIODevice::Text)) {
    QMessageBox::critical(this,
                           "QXSRExample::parseXML",
                           "Couldn't open data.xml",
                           QMessageBox::Ok);
}

QDomDocument doc("mydocument");
QDomElement docElem = doc.documentElement(); //koreňový element
//nájde potomka s názvom obývačka
QDomElement obyvaccka = docElem.firstChildElement("obyvaccka");
//z potomka obývačka načíta elementy s názvom svetlo
QDomNodeList svetla = obyvaccka.elementsByTagName("svetlo");
QDomNodeList rolety = obyvaccka.elementsByTagName("rolety");
//vyberie prvé svetlo a spraví z neho element
QDomElement svetlo = svetla.at(0).toElement();
//vyberie prvého potomka premennej svetlo
QDomNode entry_svetlo= svetlo.firstChild();
// načíta data z elementu hodnota a uloží ich do premennej
QDomElement svetloData = entry_svetlo.toElement();
QString text1 = svetloData.text(); //prevedia data do stringu
//uloží data do konečnej premennej ako číslo
obyvaccka_svetlo_1=text1.toInt();
```

Pred samotnou kompiláciou musíme ešte nastaviť cestu k nástroju „qmake-qt4“. To nastavíme v „Options“ po kliknutí na tlačítko „Tools“. Vyberieme podmenu QT4 a nastíme location „qmake-qt4“ ako vidíme na obrázku 2.20. Po tomto nastavení už stačí len kliknúť na tlačítko „Build“.



Obr. 2.20: Nastavenie cesty k nástroju QMake-Qt4.

2.4.2 Realizácia vzdialeného prístupu

Najjednoduchší vzdialený prístup je bezpochyby webové rozhranie. S podporou jazyka PHP dokážeme editovať súbory v linuxe a taktiež spúšťať skripty. Naša Qtopia 2.2.0 na SBC6410 obsahuje len webový server Boa, ktorého posledný release je z roku 1997 a PHP nepodporuje. Takže stránku budeme testovať zasa len na Ubuntu na našom notebooku. K tomu sú potrebné balíky Apache server a PHP. Ako editor použijeme BlueFish.

Cieľom je aby sme vzdialene editovali XML súbor, podľa ktorého sa nastavuje naša grafická aplikácia. Preto si vytvoríme jednoduchú webovú stránku, ktorá obsahuje PHP skripty. Najjednoduchším PHP nástrojom pre editáciu XML súborov je

nástroj SimpleXML. SimpleXML pracuje tiež s objektovým prístupom a načítaním celého súboru do pamäte. Čiže využíva DOM. Editovací formulár na našej stránke môžeme vidieť na obr.2.21.

Domček na stračej nôžke

Home Osvetlenie Teplota Rolety Izby

Kuchyňa

svetlo1: Zapnute ☐ Vypnute ☒

svetlo2: Zapnute ☒ Vypnute ☐

svetlo3: Zapnute ☒ Vypnute ☐

svetlo4: Zapnute ☐ Vypnute ☒

Roleta1 :

Roleta2 :

Roleta3 :

Hotovo

Obr. 2.21: Formulár na našej PHP stránke

Celá stránka je robená formou formulárov. Pre každý formulár máme vytvorený akčný PHP skript, ktorý sa vykoná po stlačení tlačítka „Uložiť“. Po vykonaní skriptu sa nám stránka obnoví a dáta sa načítajú z XML súboru. Akčný skript, ktorý sa vykoná po stlačení tlačítka „Uložiť“ v menu „Osvetlenie“ vyzerá takto:

```
<?php
$subor = 'data.xml';
$xml = simplexml_load_file('data.xml');
//do premennej obyvaccka sa priradi element obyvaccka
$obyvaccka = $xml->obyvaccka;
$kuchyna = $xml->kuchyna;
//do premennej svetlo1_obyvaccka sa priradi hodnota prvého svetla v
    obývačke
```

```
$svetlo1_obyvacka = $obyvacka->svetlo[0];

$svetlo1_kuchyna = $kuchyna->svetlo[0];
// načítanie hodnoty z formulára na stránke
$svetlo1_obyvacka->hodnota = $_GET['svetlo1'];

$svetlo1_kuchyna->hodnota = $_GET['svetlo1_kuchyna'];
//uloženie späť do súboru
$xml->asXML('data.xml');
$
?>
```

Aby bol tento prístup chránený, musíme zaviesť prihlasovanie sa na túto stránku. To nám umožňuje server Apache, ktorý sme si nainštalovali. Podobný postup najdeme na ich stránke[1]. Po zadaní našej IP adresy do prehliadača je teda nutné sa prihlásiť prihlasovacím menom a heslom.

2.4.3 Periférna jednotka

Našou úlohou je zostrojiť periférnu jednotku, ktorej úlohou bude merať nejakú veličinu, a s ktorou budeme schopní komunikovať. Najpodstatnejšou veličinou v obytných priestoroch je asi teplota. Preto zostrojíme teplomer. Základom tohto teplomeru bude mikroprocesor ATmega8. Naš teplomer nám bude odosielať hodnoty, ktoré odmerá, po sériovej linke RS-232.

Pre komunikáciu cez RS-232 potrebujeme prevodník úrovní. Mikroprocesor totiž pracuje s úrovňami 5 V a RS-232 pracuje s úrovňami od 5 až do 24 V. Preto potrebujeme previesť úroveň mikroprocesora aspoň na 10V. K tomu slúži obvod MAX232, ktorý na vytvorenie vyššej napäťovej úrovne využíva elektrolytické kondenzátory. Komunikácia po RS-232 si vyžaduje inicializáciu portov. Inicializácia pozostáva z nastavenia BAUD rate, nastavenia formátu rámca a povolenia vysielania a prijímania. Ak chcem využívať USART riadený prerušením, tak musí byť pri inicializácii vynulovaný „Global Interrupt Flag“.

```

#define FOSC 8000000// frekvencia oscilátora
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
void main( void )
{
    USART_Init ( MYUBRR );
}
void USART_Init( unsigned int ubrr)
{
    /* Nastavenie BAUD rate */
    UBRRH = (unsigned char)(ubrr>>8);
    UBRRL = (unsigned char)ubrr;
    /* Povolenie prijímania a vysielania */
    UCSRB = (1<<RXEN)|(1<<TXEN);
    /* Nastavenie formátu rámca: 8 data bitov a 1 stop bit */
    UCSRC = (1<<URSEL)|(0<<USBS)|(3<<UCSZ0);
}

```

Keďže integrovaný oscilátor nie je vždy najpresnejší tak potrebujeme externý oscilátor. Najlepšie taký, s ktorého frekvenciou sa dobre počíta pri nastavovaní BAUD rate. Ten nastavíme pomocou registra UBRR. Jeho hodnotu si vypočítame pomocou vzorca 2.1.

$$UBRR = \frac{f_{osc}}{16BAUD} - 1 \quad (2.1)$$

Prehľad využívaných registrov:

- UBRR – USART Baud Rate Register – Je to BAUD rate generator. Nastavíme ho na určitú hodnotu, a podľa frekvencie oscilátora sa postupne dekrementuje. A časovač pre BAUD rate je spustený zakaždým, keď UBRR dosiahne nulu.
- UCSRB – USART Control and status Register B:
 - RXEN – Receiver Enable – Zapísaním 1 do tohto bitu povolíme prijímanie znakov.

- TXEN – Transmitter Enable – Zapísaním 1 do tohto bitu povolíme odosielanie znakov.
- UCSRC – USART Control and status Register C:
 - URSEL – Riadi, do ktorého registra budeme niečo zapisovať. Ak je nastavený na 1 tak zapisujeme do UCSRC. V opačnom prípade zapisujeme do UBRRH.
 - USBS – Tu nastavujeme počet stop bitov pri vysielaní. Ak nastavíme 0 tak sa vysielajú 1 stop bit, ak nastavíme 1 vysielajú sa dva stop bity.
 - UCSZ0 – Slúži na nastavenie počtu bitov v jednom slove. Nastavujeme podľa tabuľky v datasheete [3].

Po nastavení sériového rozhrania musíme ešte nastaviť AD prevodník ATmega8 ma vstavaný 10 bitový prevodník. Ktorý je taktiež potrebné inicializovať. Prehľad registrov, ktoré je potrebné nastaviť:

- ADCSRA – ADC Control and Status Register A:
 - ADPS2, ADPS1, ADPS0 – nastavenie prescaleru podľa tabuľky v datasheete [3],
 - ADFR – po nastavení tohto bitu na 1 zapneme free running mode, ktorý umožní automatické aktualizácie,
 - ADSC – slúži na zahájenie prevodu.
- ADMUX – ADC Multiplexer Selection Register:
 - REFS0 a REFS1 – pri nastavení obidvoch bitov REFS na nulu, budeme používať ako referenčné napätie VCC,
 - ADLAR – budeme používať len 8 bitové rozlíšenie prevodníku.

Ako snímač teploty využijeme termistor. Na vstup termistoru privedieme referenčné napájacie napätie a na jeho výstup pripojíme na AD prevodník mikroprocesoru ATmega. Výstupné napätie budeme čítať z registra ADCH. Pre výpočet skutočnej teploty podľa výstupného napätia použijeme vzorec

$$T_a = \frac{V_{out} - V_{0^{\circ C}}}{T_{c1}},$$

kde

- T_a - teplota prostredia,
- V_{out} - výstupné napätie,
- $V_{0^{\circ}C}$ - výstupné napätie pri teplote $0^{\circ}C$,
- T_{c1} - teplotný koeficient.

Takto dostaneme výslednú teplotu, ktorú je možné odoslať cez sériový port. Táto hodnota je pripravená v premennej „teplota“. Nášmu teplomeru nastavíme adresu. Bude 8 bitová, keďže vysielame 8 bitové slovo. Teplomer bude stále počúvať na zbernici. Po prijatí bajtu ho porovná so svojou adresou. Ak sa zhoduje, vyšle hodnotu z premennej „teplota“. Na centrálnej jednotke je spustený skript, ktorý prijatú hodnotu zapíše do súboru. Túto hodnotu si následne prečíta grafická aplikácia, zobrazí ju, a zo súboru, z ktorého čítala ju zmaže. Tento postup je zobrazený vývojovým diagramom v prílohe A.1.

Schéma zapojenia nájdeme v prílohe B.1 a návrh plošného spoja je v prílohe C.1.

3 ZÁVER

Prvým bodom zadania bakalárskej práce bolo preštudovať súčasné koncepcie inteligentných domov a zamerať sa na ich nedostatky. Rozobrali sme a porovnali rôzne technológie, ktoré sa používajú. V inteligentných domoch sa využívajú centralizované aj decentralizované systémy. Každý systém ma svoje výhody a nevýhody. Nevýhody decentralizovaného systému sú najmä v tom, že je potrebné mať niekoľko „inteligentných“ prvkov, ktoré sa vedia rozhodovať a riešiť rôzne situácie, čo zvyšuje ich cenu. My sme sa rozhodli pre centralizovaný systém s jednou riadiacou jednotkou vo forme jednočipového počítača, ktorý ma dostatočný výkon na to, aby zvládal obsluhovať všetky periférne jednotky.

Druhým bodom zadania bolo realizovať návrh koncepcie inteligentného domu pomocou zbernice. V našej práci sme použili zbernicu RS-232, kvôli tomu, že na našom jednočipovom počítači máme RS-232 porty a nepotrebujeme žiadny prevodník. Koncepcia spočíva v tom, že komunikácia medzi zariadeniami prebieha v režime master-slave. Na požiadanie centrálnej jednotky jej periférna jednotka odošle požadované údaje.

Ďalším bodom zadania bolo realizovať grafickú aplikáciu pre jednočipový počítač. Z dôvodu jednoduchého prístupu k portom sme sa rozhodli nainštalovať do jednočipového počítača operačný systém linux. Priložená bola distribúcia QT 2.2.0, ktorá bola v našej jednotke ešte navyše upravená aby zaberala čo najmenej miesta a preto obsahovala len minimálne softvérové vybavenie. Nepodarilo sa nám ani skompilovať vývojové prostredie pre QT 2.2.0, ktoré si vyžadovalo neaktuálne knižnice. Preto sme našu grafickú aplikáciu vyvynuli v prostredí QT 4.7 s novými knižnicami. Aplikácia je jednoduchá na ovládanie a obsahuje veľké ikony, ktoré sú pohodlne ovládateľné dotykovým displejom. Aplikácia spolupracuje s databázou zariadení a ich stavov, v podobe XML súboru.

Keďže je pravdepodobné, že centrálna jednotka bude umiestnená v obytných priestoroch a bude využívaná pomerne často, tak je vhodné, aby obsahovala aj iné užitočné funkcie. Distribúcia QT 4.7 obsahuje v základnej verzii niekoľko využiteľných aplikácií. Napríklad poznámkový blok, hlasový záznamník, budík, prehrávač

hudby a pod. Všetky tieto aplikácie dokážeme spúšťať priamo z našej aplikácie.

Posledným bodom zadania bolo pripojiť k sériovému portu vhodného meracieho zariadenia. My sme skonštruovali jednoduchý teplomer na procesore ATmega8. Pripájame ho na sériový port RS232. Jeho funkcia je jednoduchá. Počúvať na sériovom porte všetku komunikáciu, a ak zaznamená svoju adresu, odošle aktuálnu teplotu.

Pre možnosť vzdialeného prístupu k informáciám o domácnosti a vzdialenému ovládaniu domu sme vytvorili webovú stránku v jazyku PHP, ktorá dokáže čítať a zapisovať do XML databázy, s ktorou spolupracuje aj grafická aplikácia. Na stránku je potrebné sa prihlasovať menom a heslom.

Naša centrálna jednotka takto dokáže spracovávať informácie od všetkých zariadení, ktoré budú schopné vyslať svoje dáta po sériovej linke. Grafická aplikácia je rozširiteľná o ďalšie ovládacie a zobrazovacie prvky, podľa toho, aké periférne jednotky k nej budú pripojené. Webovú stránku môžeme zasa rozšíriť o prístup na databázový server, kde by sa ukladali informácie o teplotách a podobne. Z týchto údajov bude schopná zobrazovať grafy, priebehy a štatistiky.

LITERATÚRA

- [1] *Apache HTTP server* [online]. 2010 [cit. 2010-12-5]. Authentication. Dostupné z URL: <<http://httpd.apache.org/docs/2.0/howto/auth.html>>.
- [2] *ARM: Architecture for the digital world* [online]. 2010 [cit. 2010-11-20]. Processors. Dostupné z URL: <<http://www.arm.com/products/processors/index.php>>.
- [3] ATmega8 Datasheet. *Atmel*[online]. 2011 [cit. 2011-05-20]. Dostupné z URL: <http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf>.
- [4] ELKO EP. *INELS : Inteligentná elektroinštalácia* [online]. 2010 [cit. 2010-11-12]. O systéme. Dostupné z URL: <<http://www.inels.sk/index.php?sekce=view>>.
- [5] GOODWIN, Steven. *Smart Home Automation with Linux*. USA : APRESS, 2010. 289 s. ISBN 978-1-4302-2779-3.
- [6] *Icon Archive* [online]. 2010 [cit. 2010-12-5]. Icons. Dostupné z URL: <<http://www.iconarchive.com/>>.
- [7] *IT-Svět.eu* [online]. 2010 [cit. 2011-23-4]. Windows 8 na ARM procesorech. Dostupné z URL: <<http://www.it-svet.eu/2011/software/news/windows-8-na-arm-procesorech/>>.
- [8] *KNX Association : [Official website]* [online]. 2010, 2010-11-06 [cit. 2010-11-11]. Introduction. Dostupné z URL: <<http://www.knx.org/knx-standard/introduction/>>.
- [9] *LonMark : Technical Resources* [online]. 2007 [cit. 2010-11-12]. Overview. Dostupné z URL: <<http://www.lonmark.org/connection/>>.
- [10] *QT 4.7* [online]. 2011 [cit. 2011-05-20]. Intro. Dostupné z URL: <<http://doc.qt.nokia.com/4.7-snapshot/qt4-7-intro.html>>.

- [11] *QT 4.7 Download* [online]. 2011 [cit. 2011-04-20]. Download. Dostupné z URL: <<http://qt.nokia.com/downloads/>>.
- [12] *RS-422 and RS-485 Application Note* [online]. Ottawa : B&B Electronics, 2006 [cit. 2010-11-13]. Dostupné z URL: <<http://www.bb-elec.com/bb-elec/literature/tech/485appnote.pdf>>.
- [13] *ZigBee Alliance* [online]. 2010 [cit. 2010-11-16]. Home. Dostupné z URL: <<http://www.zigbee.org/Home.aspx>>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

Bd BAUD- jednotka modulačnej rýchlosti

°C stupeň celzia

CIB Common Installation Bus

DCSK Differential Code Shift Keying

DDR Double Data Rate- dvojnásobná rýchlosť spracovania dát

DMA Direct Memory Access - je to kanál s priamim prístupom k pamäti.

DOM Document Object Model - metóda pri parsovaní XML

EHS Electronic Home System

EIB European Installation Bus

Ethernet Technológia pre káblový prenos dát.

IrDA Infrared Data Association - prenos dát infračerveným svetlom

joystick Páčka ktorá slúži k ovládaniu kurzoru na obrazovke

MHz MegaHertz

NAND Not And - logický negovaný súčin

OFDM Orthogonal frequency-division multiplexing

PLC Power Line Communication

SAX Simple API for XML - metóda pri parsovaní XML

SBC Single Board Computer

SD Secure Digital

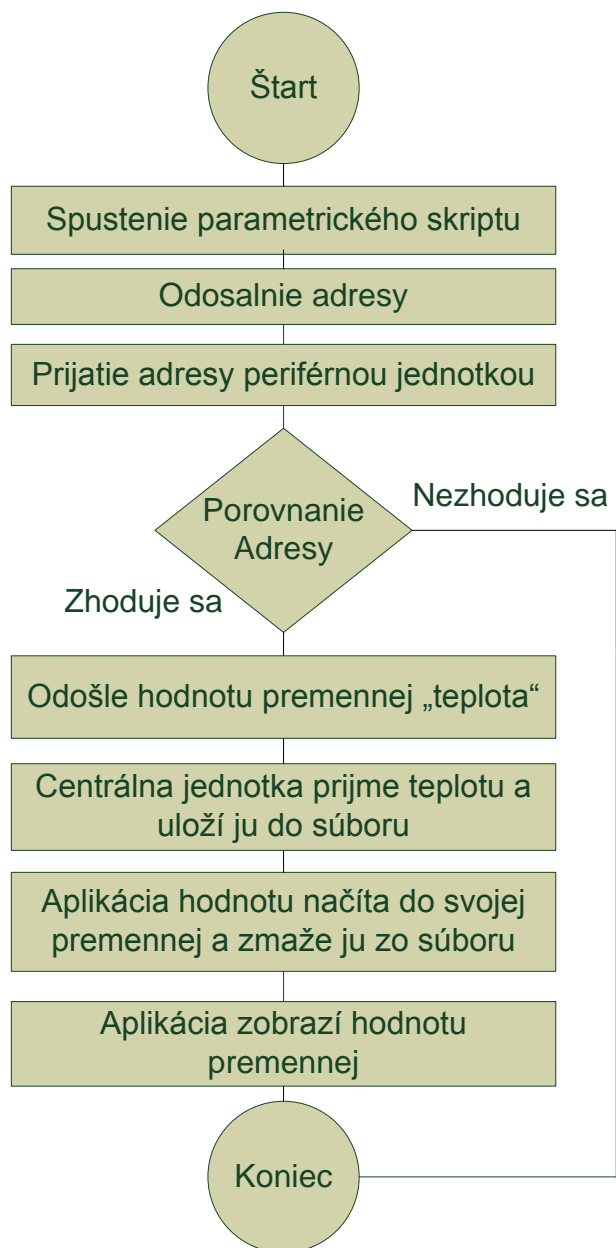
TFT-LCD Thin film transistor liquid crystal display - displej s tekutých kryštálov,
ktorý pre zlepšenie kvality obrazu využíva tranzistory s jednou vrstvou
vyvedenou mimo tranzistor

WiFi Bezdrôtová technológia pre prenos dát

XML Extensible Markup Language - značkový jazyk

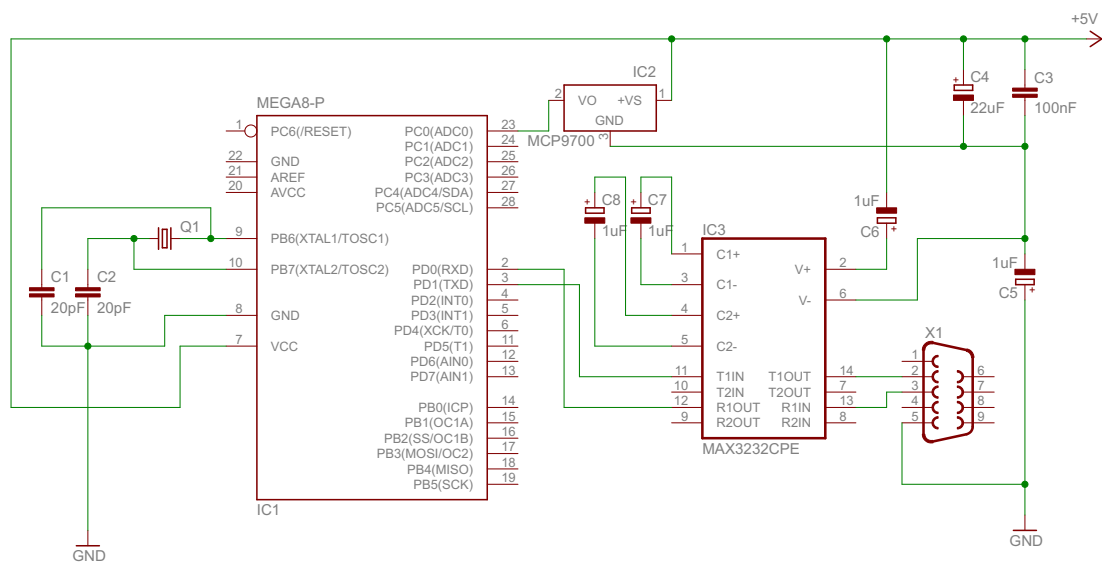
ZigBee Bezdrôtová technológia pre prenos dát v sietach PAN (do 75m) Určená pre riadenie kvôli nízkej rýchlosti prenosu dát.

A DIAGRAM UDALOSTÍ



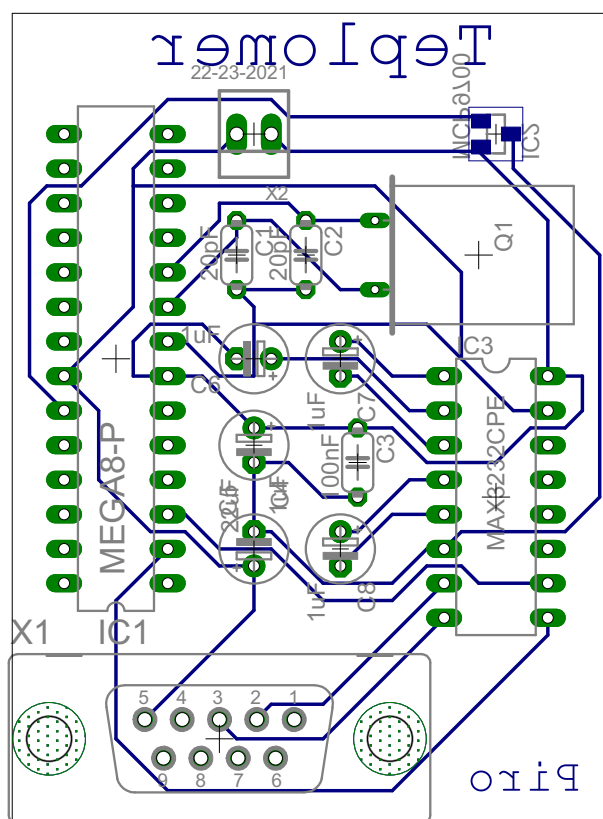
Obr. A.1: Diagram udalostí po stlačení akčného tlačítka.

B SCHÉMA ZAPOJENIA TEPLOMERA



Obr. B.1: Schéma zapojenia teplomera.

C SCHÉMA PLOŠNÉHO SPOJA TEPLOMERA



Obr. C.1: Doska plošného spoja teplomera.